9. OWASP Top 10

🔐 Web Application Security & OWASP Top 10

1. What is the OWASP Top 10?

- The **OWASP Top 10** is a globally recognized list of the **most critical web application security risks**, updated periodically by the Open Web Application Security Project (OWASP).
- It serves as a baseline for developers, security pros, and organizations to prioritize security efforts.
- Categories include injection, broken authentication, XSS, misconfigurations, and more.
- Provides **awareness, mitigation strategies, and best practices**.

2. Why is Injection Dangerous?

- Injection attacks occur when untrusted data is sent to an interpreter as part of a command or query (e.g., SQL, OS commands).
- Attackers can manipulate input to execute unintended commands, leading to **data theft, data corruption, or full system compromise**.
- Examples: SQL Injection, Command Injection, LDAP Injection.
- Often easy to exploit and have severe consequences, making it one of the highest risks.

3. How Does XSS Affect Web Applications?

- **Cross-Site Scripting (XSS)** allows attackers to inject malicious scripts into trusted websites.
- Victims who visit the compromised pages unknowingly execute the attacker's script in their browsers.
- Consequences: session hijacking, defacement, phishing, spreading malware.
- Three types: Stored XSS, Reflected XSS, and DOM-based XSS.

4. What is the Risk of Broken Authentication?

- Broken authentication allows attackers to **compromise user credentials or session tokens**.
- Leads to unauthorized access, impersonation, and potentially full control over user accounts or even admin privileges.
- Causes include weak password policies, poor session management, or leaked credentials.

5. Can You Explain Sensitive Data Exposure?

- Occurs when sensitive info (passwords, credit cards, personal data) is **not properly protected** in storage or transit.

- Risks include data breaches, identity theft, financial loss.
- Happens due to lack of encryption, weak cryptographic algorithms, or improper key management.

---

6. Describe a Security Misconfiguration

- Security misconfiguration means **insecure default settings, incomplete configurations, or ad-hoc fixes** that leave systems exposed.
- Examples: default passwords, verbose error messages, unnecessary services running, open cloud storage.
- Can be exploited to gain unauthorized access or info.

---

7. What is XML External Entity (XXE)?

- XXE is an attack against XML parsers that process **external entities**.
- Attackers craft XML input referencing external resources, potentially exposing sensitive files, causing DoS, or SSRF attacks.
- Mostly affects older or poorly configured XML processors.

---

8. How Do Broken Access Controls Impact Security?

- Broken access controls allow attackers to **bypass authorization checks**, accessing resources or functions they shouldn't.
- Can lead to data leaks, privilege escalation, or manipulation of data.
- Common causes: missing checks, insecure direct object references (IDOR).

---

9. What Are Common Web Application Security Flaws?

- Injection vulnerabilities
- Broken authentication and session management
- Cross-Site Scripting (XSS)
- Insecure Direct Object References (IDOR)
- Security misconfiguration
- Sensitive data exposure
- Cross-Site Request Forgery (CSRF)
- Insufficient logging and monitoring
- Using components with known vulnerabilities

---

10. How to Prevent Insecure Deserialization?

- Avoid deserializing data from untrusted sources.
- Use strict type checking and integrity verification on serialized data.
- Implement **whitelisting** for allowed classes or types.

- Use safe serialization formats (e.g., JSON over native serialization).
- Apply runtime protections and monitoring for suspicious deserialization behavior.

---

11. What is the Use of Security Logging and Monitoring?

- Enables detection of **suspicious or malicious activities** in real-time or post-incident.
- Helps in forensic investigations, compliance, and incident response.
- Logs should be **tamper-proof, comprehensive, and timely** analyzed to identify attacks early.
- Lack of it delays breach detection and increases damage.

---

12. Explain the Risks of Using Components with Known Vulnerabilities

- Using outdated or vulnerable third-party libraries/frameworks exposes your app to **known exploits**.
- Attackers exploit these weaknesses to gain unauthorized access or execute arbitrary code.
- Requires regular updates, vulnerability scanning, and dependency management.

---

13. How Can Using APIs Increase Security Risks?

- APIs often expose critical business logic and data.
- Poorly secured APIs can lead to data leaks, unauthorized actions, and privilege escalation.
- Risks include broken authentication, lack of rate limiting, injection vulnerabilities, and excessive data exposure.
- API security requires strict access control, input validation, encryption, and monitoring.
